IFW AF/2171

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of	
John Wong	Group Art Unit: 2171
Application No.: 09/810,246	Examiner: Sana A Al Hashemi
Filed: March 19, 2001	Confirmation No.: 8377
For: MIRROR FILE SYSTEM	

BRIEF FOR APPELLANT

Mail Stop APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This appeal is from the decision of the Primary Examiner dated March 16, 2004, finally rejecting claims 1-12, 14 and 15, which are reproduced as an Appendix to this brief.

A check covering the requisite fee under 37 CFR 41.20(b)(2) accompanies this Brief. The Commissioner is hereby authorized to charge any fees that may be required by this paper, and to credit any overpayment, to Deposit Account No. 02-4800.

Table of Contents

Page

Real Party in Interest 1
Related Appeals and Interferences 1
Status of Claims 1
Status of Amendments 1
Summary of Claimed Subject Matter 2
Grounds of Rejection to Be Reviewed on Appeal 3
Argument 4
A. Claims 1-10 4
B. Claim 2 6
C. Claim 5 7
D. Claim 6 7
E. Claims 11, 12, 14 and 15 7
F. Claims 14 and 15 8
Conclusion 10
APPENDIX A - The Appealed Claims A-1
APPENDIX B - Evidence B-1
APPENDIX C – Related Proceedings C-1

Real Party in Interest

The subject application is assigned to Twin Peaks Software, Inc., a California corporation.

Related Appeals and Interferences

There are no other prior or pending appeals, interferences or judicial proceedings known to Appellant, Appellant's legal representative, or the assignee, which may be related to, directly affect, or be directly affected by, or have a bearing on, the Board's decision in this appeal.

Status of Claims

The application contains claims 1-16, all of which are currently pending. Claims 13 and 16 have been identified as containing allowable subject matter. The remaining claims, namely claims 1-12, 14 and 15, stand finally rejected, and form the basis for this appeal.

Status of Amendments

There were no amendments filed subsequent to the final Office Action.

Summary of Claimed Subject Matter

The claimed subject matter is directed to file systems for computers, and more particularly to a virtual file system that links two or more physical file systems together, and mirrors between them in real time. The structure and operation of the invention can be understood with reference to Figures 3 and 4 of the application. Figure 3 illustrates two physical file systems A and B. As is conventional, each file system contains a number of directories and files that are organized in a hierarchical manner. For example, file system A contains a root directory /A and two subdirectories b and i at a first level under the root directory. Further directories and/or files are arranged at lower levels of the hierarchy. In the example of Figures 3 and 4, a portion of the structure of file system A is to be linked to file system B. This linked structure comprises the directory b and each of the directories or files c-h that are nested under directory b. This structure, labeled 220, is to be linked to the structure 221 that includes the nested directories and/or files y, z and v in file system B. (Page 19, lines 3-6).

Referring to Figure 4, the directory y of file system B is mounted onto the directory b of file system A. As a result, it can be seen in Figure 4 that the directory b now contains, in addition to subdirectories c-h, the subdirectories z and v that were contained in directory y of file system B. In addition, the directory b of file system A is mounted onto the directory y of file system B. As illustrated in Figure 4, directory y of file system B therefore contains subdirectories c-h, as well as subdirectories z and v. (Page 19, lines 10-15).

A virtual file system 203 is created in conjunction with these mounting operations. This virtual file system has a data structure that contains all of the elements of the structure 220 of file system A and the structure 221 of file system B. Thus, it has a root directory b/y 231, and subdirectories or files c-h, z and v. (Page 19, line 19 to page 20, line 11).

Every element of a file system, namely each file and each directory, has a data structure known as a *vnode* that contains information and the operations that can be performed on the file or directory. In the virtual file system disclosed in the application, every file or directory in the virtual file system 203 has a super *vnode* structure that is called an *mnode*. This *mnode* contains a *vnode* structure and two *vnode* pointers. The *vnode* structure comprises the *vnode* for the given file or directory within the virtual file system 203, and the two *vnode* pointers respectively point to the actual *vnode* of the

corresponding file or directory within the two physical file systems A and B. The two *vnode* pointers are represented in Figure 4 by the small black squares within the directories 231 to 233, 236 and 238, with arrows that point to their corresponding structures in the file systems A and B. (Page 21, lines 5-16; page 21, line 29 to page 22, line 7).

In operation, when a request is made to access a file or directory under the b directory of file system A or the y directory of file system B, the file system detects that these directories have the virtual file system 203 mounted on them. As a result, all file access requests are directed to the *vnode* for the virtual directory b/y of the virtual file system 203. Using the *mnode* data structure, the system obtains the *vnodes* for both the directory b in file system A and directory y in file system B. The requested operation, e.g. a write operation, is then sent to both of the vnodes. As a result, real time mirroring is effected between the corresponding pairs of elements, i.e. files or directories, in the two physical file systems A and B. (Page 22, line 8 to page 23, line 32).

A mirror file system of this type can be configured in a number of different ways, as depicted in Figures 5-10 of the application. Exemplary Figure 8 illustrates a client system 700 that uses a mirror file system Z to access two physical file systems A and B, for example on different servers 710 and 720. In this configuration, file system A and file system B are exported to the client, where they are mounted by the virtual file system Z. In this configuration, operations performed by the client upon elements stored in the file system of either server are automatically mirrored to the file system of the other server, in a real-time fashion. (Page 29, lines 1-22).

Grounds of Rejection to Be Reviewed on Appeal

The final Office Action sets forth two grounds of rejection that are to be reviewed on this appeal:

1. Are claims 1-10 unpatentable under 35 U.S.C. §103 when the Kramer patent (U.S. 6,216,140) is considered in view of the Skiba patent (U.S. 6,366,988)?

2. Are claims 11, 12, 14 and 15 unpatentable under 35 U.S.C. §103 when the Skiba patent is considered in view of the Kramer patent?

Argument

A. Claims 1-10

The rejection of claims 1-10 alleges that the Kramer patent discloses the claimed concept of mounting components of each of two physical file systems in a single directory. The Kramer patent is particularly directed to software development. As described in column 4, lines 42-51, after a particular version of software has been developed and released, changes continue to be made to the software. To accommodate these changes, the Kramer patent discloses a technique in which a hierarchy of the files for the released version is created and copied, so that all changes that are made for a subsequent version are isolated from the first version.

The Kramer patent is particularly concerned with the ability to make a copy of hierarchically organized information without unnecessarily utilizing resources. To do this, the Kramer patent discloses a technique in which a virtual copy of a hierarchy of files is first created, simply by adding a parent link to the root of a shared directory hierarchy (column 5, lines 21-23). Then, when a modification is to be made to a portion of the hierarchy of files, an actual copy of that portion is made (column 6, lines 1-15).

The Kramer patent also discloses that, in some aspects of software development, a new feature is developed independently of other features, and then merged with the other features if it is found to be acceptable (column 9, lines 22-26). The discussion at columns 9-12 explains how such a merger takes place within the context of the Kramer system.

The rejection of claims 1-3 and 7-9 alleges that the Kramer patent discloses "mounting components of each of ... two physical file systems in a single directory," with reference to the objects of the invention described in column 3, lines 16-19 and 36-38, and the merge operation described in column 9, lines 51-54 and column 13, lines 38-40. However, none of these portions of the Kramer patent, nor any other portions thereof, pertain to the mounting of file systems, particularly two different physical file systems. It is important to note that the operations described in the Kramer patent all take place within the context of a *single* file system. There is no disclosure in the Kramer patent pertaining to multiple file systems, much less the mounting of components of each of two physical file systems in a single directory, as recited in claims 1 and 4.

In responding to Appellant's argument to this effect, the final Office Action relies upon an overly broad definition of the term "file system." At page 8, paragraph 10.a, the Office Action states, "The Examiner is interpreting [the term 'file systems'] to be as the 'directories'; and the 'components' of these file systems to be the 'files' of the directories." The Office Action does not identify any support for such an interpretation of the term "file system."

A copy of a relevant page from the Microsoft Computer Dictionary, 5th Edition, was submitted with the response to the final Office Action, and is attached hereto in Appendix B. As indicated therein, the term "file system" is defined as "the overall structure in which files are named, stored and organized. A file system consists of files, directories, or folders, *and* the information needed to locate and access these items." (emphasis added).

From this definition, it can be seen that a file system is not equivalent to a directory, per se, as alleged in the final Office Action. Rather, a directory is merely one component of a file system. In addition to this component, the file system also includes the software module that organizes and manipulates the structure and location of files on the physical storage media, e.g., disk drives. Typically, this module is located between the actual data stored on the drives and user applications, to perform operations such as reading, writing and storing the data.

The Court of Appeals for the Federal Circuit recently confirmed that claims are to be construed from the vantage point of a person skilled in the relevant art. *Vanderlande Industries Nederland B.V. vs. ITC*, 70 USPQ2d 1696, 1704 (Fed. Cir. 2004). The court indicated that technical dictionaries are the types of evidence that demonstrate the special meanings to apply to terms of art that are used in the claims. As illustrated by the accompanying definition from the Microsoft Computer Dictionary, a person of ordinary skill in the art would not consider a "directory" to be the same as a "file system". Since the rejection of the claims is based upon such an erroneous interpretation, it cannot be affirmed.

Specifically, as discussed above, the Kramer patent only discloses the merger of two or more hierarchies of files and directories within a *single* file system. Since directories may be components of file system data entities, but are not themselves file systems, Kramer's teaching relating to the merging of two or more hierarchies of files and

directories does not, in any way, suggest the mounting of plural physical file systems in a directory.

The Office Action indicates that the Kramer patent does not disclose elements in a virtual file system data structure with two associated pointers that respectively point to application interface data structures of corresponding components in each of two physical file systems. To this end, therefore, the rejection relies upon the Skiba patent, with particular reference to column 2, lines 48-54. This portion of the patent discloses the creation of an expanded volume, in which subfolders point to folders on a different volume in a network. This vague reference to pointers does not, in any way, disclose nor otherwise suggest the concept of utilizing two associated pointers in elements of a virtual file system data structure, to point to application interface data structures of corresponding components in each of two physical file systems.

Like the Kramer patent, the Skiba patent also pertains to operations that take place within a single file system. The portion of this patent that was specifically identified in the Office Action, namely column 2, lines 48-54, discusses the Microsoft Distributed File System (DFS). In this file system, a logical volume (or storage unit) can be distributed across multiple physical volumes. However, each of these physical volumes is a part of the same file system, namely the Microsoft DFS. The Skiba patent does not suggest that a subfolder in a logical volume points to a folder that is in a *different* file system.

Consequently, even if the teachings of the Skiba patent relating to the distributed file system were to be employed within the software development environment of the Kramer patent to provide for expanded storage capacity, the resulting arrangement would still be contained within a single file system. There is no suggestion in either patent, or in their combined teachings, to mount components of two physical file systems within a single directory, or otherwise perform operations across two different file systems.

B. Claim 2

Claim 2 recites that the application interface data structures contained within the components of each of the two physical file systems and the virtual file system correspond to a vnode structure. Appellant is unable to find any discussion of a vnode structure in either of the two applied references. Furthermore, the final Office Action does not address

the subject matter of claim 2, nor otherwise provide any support for the rejection of this claim.

There simply is no basis to assert that the subject matter of claim 2 is suggested by the references, whether considered individually or in combination.

C. Claim 5

Claim 5 recites that a request to perform a write operation on one of the components of a physical file system designates that component by means of a path name that is common to both of the physical file systems. In rejecting this claim, the final Office Action refers to the Skiba patent at column 16, lines 30-32, which states that a file request contains the full file name. However, there is nothing in this statement which suggests that this full file name is common to two physical file systems. Rather, the operation described in the referenced portion of the patent takes place within the context of a single file system.

D. Claim 6

Claim 6 recites that the step of performing a write operation includes acquiring a lock for each copy of a component, and inhibiting the write operation until both locks can be acquired. The rejection of claim 6 refers to the Skiba patent at column 15, lines 47-50, which states that the mirroring driver processes numerous file requests, which include Lock/Unlock. However, this vague reference to Lock/Unlock requests is not sufficient to disclose, nor otherwise suggest, the claimed subject matter. Specifically, the claim recites that a lock is acquired for *each* of two copies of a component, and the write operation is not performed until *both* locks have been acquired. The Skiba patent merely acknowledges that Lock/Unlock operations are performed, but does not disclose how they are performed within the mirroring process. There is no disclosure suggesting the use of two locks that are respectively associated with each of two copies of a component.

E. Claims 11, 12, 14 and 15

Claim 11 recites a mirrored file system that comprises a first server having a first local file system and a first physical storage device, and a second server having a second local file system and a second physical device. Claims 11, 12, 14 and 15 were rejected as

being unpatentable over the Skiba patent in view of the Kramer patent. In rejecting claim 11, the final Office Action refers to the Skiba patent at column 13, lines 59-62, and column 17, lines 5-12, as allegedly disclosing these claimed features. However, these two portions of the Skiba patent merely make general reference to a server, or servers, on a network. They do not disclose that each of two servers has its own file system and physical storage device associated therewith.

Claim 11 goes on to recite a client device having a virtual file system which mounts an imported file system from the first server and an imported file system from the second server, to provide a single point of access for components stored in each of the first and second local file systems. The rejection set forth in the final Office Action acknowledges that the Skiba patent does not disclose this claimed subject matter, and therefore relies upon the Kramer patent as suggesting such.

As discussed previously, the Kramer patent does not contain any disclosure relating to multiple file systems. Rather, it only discloses the copying of a hierarchy of files within the context of a single file system. As such, it cannot be interpreted to disclose a client device having a virtual file system, which mounts imported file systems from each of two servers. Nor does it disclose the concept of a client which acts as a single point of access for components stored in each of two file systems.

Furthermore, the Kramer patent does not contain any disclosure pertaining to the *importing* of one or more file systems into another file system. Since the Kramer patent is only concerned with operations that take place within a single file system, there is no reason to import the file system of a server into the file system of a client device.

Consequently, even if the teachings of the Kramer patent could be applied to the system of the Skiba patent, the result would not be the same as the subject matter recited in claim 11.

F. Claims 14 and 15

Claim 14 recites that each of the first and second servers includes a virtual file system that mounts components of that server's local file system and components of the other server's local file system in a single directory. Claim 15 recites that it is these virtual file systems that are imported into the client. The final Office Action does not address the subject matter of either of these claims. Neither reference contains any disclosure of a

server having a virtual file system that mounts components of its own local file system and the file system of another server, nor the importation of such virtual file systems into a client device. The final Office Action does not provide any support for the rejection of these claims.

Conclusion

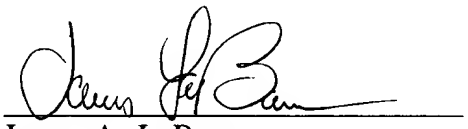
In summary, the rejection of the claims is based upon an improper interpretation of the term "file system" to read upon individual directories. The Examiner has failed to provide any support for his position in light of the contradictory evidence provided by Appellant.

Furthermore, even if the broad interpretation of "file system" is accepted, the final Office Action still fails to identify where a number of claimed features are suggested by the references, either individually or in combination.

The rejections of the claims are not properly founded in the statute, and should be reversed.

Respectfully submitted,
Burns, Doane, Swecker & Mathis, L.L.P.

Date: October 18, 2004

By: 
James A. LaBarre
Registration No. 28,632

P.O. Box 1404
Alexandria, Virginia 22313-1404
(703) 836-6620

APPENDIX A - The Appealed Claims

1. A virtual file system which provides mirroring and linking of two physical file systems, comprising:

means for mounting components of each of said two physical file systems in a single directory; and

a virtual file system data structure containing elements which respectively correspond to each of the mounted components, each of said elements having an application interface data structure with two associated pointers that respectively point to application interface data structures of a corresponding component in each of said two physical file systems.

2. The virtual file system of claim 1, wherein said application interface data structures correspond to a vnode structure.

3. The virtual file system of claim 1, wherein said components comprise directories and files.

4. A method for sharing files in a computer system, comprising the steps of:

mounting components of each of two physical file systems in a single directory, such that a copy of each component is stored in each of said two physical file systems;

receiving a request to perform a write operation on one of said components; and

performing said write operation on both copies of said one component in said two physical file systems, respectively, in real time in response to said request.

5. The method of claim 4 wherein said request designates said one component, on which the write operation is to be performed, by means of a path name that is common to both of said physical file systems.

6. The method of claim 4 wherein the steps of performing said write operation includes the steps of acquiring a lock for each copy of said one component, and inhibiting said write operation until both locks can be acquired.

7. The virtual file system of claim 1, wherein said mounting means mounts a directory of one of said physical file systems to a directory of the other physical file system.

8. The virtual file system of claim 1 wherein said single directory functions as a single mount point for access to the components of either of said two physical file systems.

9. The virtual file system of claim 1 wherein the mounted components of each physical file system are replicated in the other physical file system.

10. The method of claim 4 wherein said mounting step comprises mounting a directory of one of said physical file systems to a directory of the other physical file system.

11. A mirrored file system, comprising:

a first server having a first local file system and a first physical storage device associated therewith;

a second server having a second local file system and a second physical storage device associated therewith; and

a client device having a virtual file system which mounts an imported file system from said first server and an imported file system from said second server to provide a single point of access for components stored in each of said first and second local file systems.

12. The mirrored file system of clam 11, wherein said first local file system and said second local file system are each imported into said client device, and said virtual file system mounts components of each of said two imported file systems to a single directory.

14. The mirrored file system of claim 11 wherein each of said first and second servers includes a virtual file system that mounts components of said server's local file system and components of the other server's local file system in a single directory.

15. The mirrored file system of claim 14 wherein the file systems that are imported into said client device comprise the virtual file systems of said first and second servers.

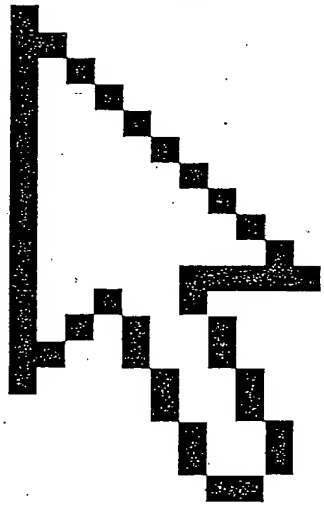
APPENDIX B - Evidence

Microsoft Computer Dictionary, Fifth Edition, 2002, page 213

Microsoft

Microsoft Computer Dictionary

Fifth Edition



PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Library of Congress Cataloging-in-Publication Data
Microsoft Computer Dictionary.--5th ed.
p. ; cm.
ISBN 0-7356-1495-4
1. Computers--Dictionaries. 2. Microcomputers--Dictionaries.

AQ76.5. M52267 2002
004'.03--dc21 200219714

Printed and bound in the United States of America.

1 2 3 4 5 6 7 8 9 QWT 7 6 5 4 3 2

Distributed in Canada by Penguin Books Canada Limited.

A CIP catalogue record for this book is available from the British Library.

Microsoft Press books are available through booksellers and distributors worldwide. For further information about international editions, contact your local Microsoft Corporation office or contact Microsoft Press International directly at fax (425) 936-7329. Visit our Web site at www.microsoft.com/mspress. Send comments to *mspinput@microsoft.com*.

Acquisitions Editor: Alex Blanton
Project Editor: Sandra Haynes

Body Part No. X08-41929

work users request files and make changes to them. To deal with the tasks of handling multiple—sometimes simultaneous—requests for files, a file server contains a processor and controlling software as well as a disk drive for storage. On local area networks, a file server is often a computer with a large hard disk that is dedicated only to the task of managing shared files. *Compare* disk server.

File Server for Macintosh *n.* An AppleTalk network integration service that allows Macintosh clients and personal computers clients to share files. *Also called:* MacFile. *See also* Print Server for Macintosh, Services for Macintosh.

file sharing *n.* The use of computer files on networks, wherein files are stored on a central computer or a server and are requested, reviewed, and modified by more than one individual. When a file is used with different programs or different computers, file sharing can require conversion to a mutually acceptable format. When a single file is shared by many people, access can be regulated through such means as password protection, security clearances, or file locking to prohibit changes to a file by more than one person at a time.

file size *n.* The length of a file, typically given in bytes. A computer file stored on disk actually has two file sizes, logical size and physical size. The logical file size corresponds to the file's actual size—the number of bytes it contains. The physical size refers to the amount of storage space allotted to the file on disk. Because space is set aside for a file in blocks of bytes, the last characters in the file might not completely fill the block (allocation unit) reserved for them. When this happens, the physical size is larger than the logical size of the file.

filespec *n.* *See* file specification (definition 1).

file specification *n.* **1.** The path to a file, from a disk drive through a chain of directory files to the file name that serves to locate a particular file. Abbreviated filespec. **2.** A file name containing wildcard characters that indicate which files among a group of similarly named files are requested. **3.** A document that describes the organization of data within a file.

file structure *n.* A description of a file or group of files that are to be treated together for some purpose. Such a description includes file layout and location for each file under consideration.

file system *n.* In an operating system, the overall structure in which files are named, stored, and organized. A file system consists of files, directories, or folders, and the information needed to locate and access these items. The term can also refer to the portion of an operating system that translates requests for file operations from an application program into low-level, sector-oriented tasks that can be understood by the drivers controlling the disk drives. *See also* driver.

file transfer *n.* The process of moving or transmitting a file from one location to another, as between two programs or over a network.

File Transfer Protocol *n.* *See* FTP[1] (definition 1).

file type *n.* A designation of the operational or structural characteristics of a file. A file's type is often identified in the file name, usually in the file name extension. *See also* file format.

fill[1] *n.* In computer graphics, the colored or patterned "paint" inside an enclosed figure, such as a circle. The portion of the shape that can be colored or patterned is the fill area. Drawing programs commonly offer tools for creating filled or nonfilled shapes; the user can specify color or pattern.

fill[2] *vb.* To add color or a pattern to the enclosed portion of a circle or other shape.

fill handle *n.* The small black square in the lower-right corner of a cell selection. When you point to the fill handle, the pointer changes to a black cross.

film at 11 *n.* A phrase sometimes seen in newsgroups. An allusion to a brief newsbreak on TV that refers to a top news story that will be covered in full on the 11 o'clock news, it is used sarcastically to ridicule a previous article's lack of timeliness or newsworthiness. *See also* newsgroup.

film recorder *n.* A device for capturing on 35-mm film the images displayed on a computer screen.

film ribbon *n.* *See* carbon ribbon.

filter *n.* **1.** A program or set of features within a program that reads its standard or designated input, transforms the input in some desired way, and then writes the output to its standard or designated output destination. A database filter, for example, might flag information of a certain age. **2.** In communications and electronics, hardware or software that selectively passes certain elements of a signal and eliminates or minimizes others. A filter on a communications network, for example, must be designed to transmit a certain frequency but attenuate (dampen) frequencies above it (a lowpass filter), those below it (a highpass filter), or those above and below it (a bandpass filter). **3.** A pattern or mask through which data is passed to weed out specified items. For instance, a filter used in e-mail or in retrieving newsgroup messages can allow users to filter

APPENDIX C – Related Proceedings

(none)